

## Práctico III

### Strings

1. El usuario podrá ingresar nombre y apellido. El programa deberá convertir las iniciales en mayúsculas y las demás letras en minúsculas.
2. Se le permitirá al usuario ingresar una frase. Se mostrarán en pantalla solamente las letras en posiciones pares de la misma.
3. Diseñar un algoritmo en pseudocódigo que permita ingresar una frase al usuario y una letra, y determine cuántas veces está esa letra en dicha frase. Luego que ya tenga el pseudocódigo, implementarlo en Python.
4. Ingresar una frase que contenga símbolos varios, además de mayúsculas y minúsculas mezclados. Determinar la cantidad de espacios, y cada símbolo que hay en la misma.
5. Permita al usuario ingresar una frase. Cambie las letras *a* por 4 y las letras *e* por 3.
6. Permita al usuario ingresar el nombre de un archivo, genere un nuevo nombre donde los espacios sean reemplazados por guión bajo y la extensión por numerales.
7. Permitir ingresar al usuario un número de un dígito. Controlando se haya ingresado dicho número de no más de 1 dígito de longitud, pasarlo a letras y mostrarlo en pantalla. (*Ejemplo: Si ingresa 3, se verá como resultado "tres"*).
8. Se le pedirá al usuario una frase. Se mostrarán en pantalla, una palabra por línea de la misma. *\*no usar listas en este ejercicio*

### Listas

9. Pedir el nombre al usuario, y corroborar si ese nombre existe entre los nombres de usuarios válidos guardados en una lista.
10. Implemente el programa que pide la usuario 8 nombres del algoritmo del práctico anterior. En ese ejercicio tenía que intentar diseñar un algoritmo que seleccionase los nombres que empiezan con la letra M de una serie de nombres otorgados por el usuario. Utilice para resolverlo los tipos de datos y comandos que le parezcan más apropiados.
11. Realizar un programita que le pida ingresar una frase al usuario y coloque cada palabra de la misma como elemento de una lista.
12. Realizar un programita que le pida ingresar una frase al usuario y coloque cada letra como elemento de una lista.
13. El usuario deberá poder ingresar varios nombres completos (*ejemplo: "Luis Pérez"*). El programa deberá luego, colocar los nombres en una lista y los apellidos en otra.
14. Se deberán ingresar 8 notas. Se mostrará el promedio, redondeado a 2 decimales.

15. Pedir al usuario una frase. Determinar de al menos dos modos diferentes (*con y sin listas*) la cantidad de palabras que hay en dicha frase.
16. Dada una lista de números, ingresada por el usuario o inventada por usted, cree otra lista con la cantidad de dígitos de cada número de la misma.

## Pseudo Aleatoriedad

17. Tirar 20 veces un dado de 6 caras. Mostrar el promedio de esas 20 tiradas.
18. Tirar ahora, 2500 veces un dado de 6 caras. Mostrar el promedio de esas tiradas. Comparar con el promedio del ejercicio anterior. ¿Nota una diferencia sustancial habiendo cambiado la cantidad de tiradas?
19. Pedirle al usuario sus 10 marcas favoritas. Mostrar una marca al azar de la lista.

## Variado

20. Pedirle al usuario la cantidad de notas que desea ingresar. Luego pedir cada nota, y guardarlas.
21. Dado un  $n$  ingresado por el usuario, realizar la suma de los  $n$  primeros términos de la serie a continuación. Mostrar el resultado.

$$\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$$

22. Cuenta Regresiva: Se requiere un programa que permita el ingreso de un número positivo y muestre en pantalla la cuenta regresiva desde el número ingresado hasta llegar a 0. Realizar diferentes versiones del programa, utilizando en cada una, una estructura de bucle diferente de las que tiene disponibles en Python.
23. Suma de Números Positivos y Negativos: Se requiere un programa que permita el ingreso de 10 números y al finalizar muestre en pantalla la cantidad números positivos y por otra parte la cantidad de números negativos que fueron ingresados.
24. Permitir ingresar números enteros hasta que se ingrese la opción "s" de salir. Plantee primero el diseño en pseudocódigo del algoritmo e implemente luego dos versiones.
  - a) La primer implementación funcionaría en cualquier lenguaje.
  - b) La segunda implementación deberá aprovechar que Python es un lenguaje dinámicamente tipado.
25. Diseñar e implementar un algoritmo que permita ingresar una serie de números, sumar todos los pares y al terminar la serie mostrar dicha suma. Si se ingresó algún impar, mostrar un mensaje *Se ingresaron impares*. Para finalizar el ingreso, indicar la cantidad de números a ingresar al principio del programa, o interrumpir la carga cuando se ingrese el número 99.
26. Permitir ingresar 10 números al usuario. Determinar y mostrar el menor y el mayor.

27. Número Invertido: Se requiere mostrar en pantalla un número invertido de 6 cifras, al que fuera ingresado por teclado. (*Ejemplo: en pantalla se verá: “El número ingresado es 140975, invertido es: 579041”*)
28. Pedirle al usuario dos números positivos,  $a$  y  $b$ . Controlar que  $a < b$ . Mostrar en pantalla los números del intervalo cerrado  $[a, b]$  La computadora deberá ahora seleccionar al azar un número de ese intervalo. Y el usuario deberá adivinar cuál número ha sido seleccionado por la computadora, obteniendo un mensaje de éxito en caso de acertar. El usuario solo tendrá 10 vidas (*número de intentos*) y en caso de no acertar, deberá obtener un mensaje de pucha.
29. El problema es el siguiente, el usuario deberá poder ingresar la longitud de la base de una pirámide y el algoritmo deberá imprimir en pantalla una pirámide de numerales. Por ejemplo, si se ingresa 7, se debería ver en pantalla:
- Diseñe el algoritmo que imprima ese triángulo, en pseudocódigo.
  - Determine que restricciones debería contemplar para que el triángulo quede bien formado. ¿Cualquier valor para la longitud de la base servirá?
  - Implemente el programa en Python a partir del pseudocódigo creado.

---

```
#  
###  
#####  
#####
```

---